A picture containing text

Description automatically generated

**Assessment**

**Software Development.**

# Unit 1 Project: Introduction to Software Development

### Project Overview:

In weeks 1-3, you will be gaining an introduction to software development – what it is, and why we do it. This project checks your understanding of the purpose of software development, the steps you take to write software code, who does what, and your ability to write basic code following a logical approach.

### Project Scenario:

You have just joined a software development company as a trainee. The company makes software for educational establishments (schools, colleges and universities). As a new trainee, you have been given an introductory task – to create a simple calculator software program.

This software will eventually be used by primary school children (aged 4-11) to help them accurately learn their basic arithmetic skills (addition, subtraction, multiplication, and division). It will also be a way for you to showcase your abilities to your new company and give them confidence in your ability to conduct all aspects of software development! You will have free reign to decide how the software will be designed and implemented, but it is suggested that the software is coded in a popular programming language.

## Task 1 (Week 1)

Create a new word-processed document. This document will contain evidence for all three tasks related to this project. Start by adding the following as an introduction:

1. Briefly explain in your own words the purpose of the project, and the key requirements that will need to be fulfilled.

|  |
| --- |
|  |

1. Explain:
   1. What the stages of the SDLC are, and what they involve generally.

|  |
| --- |
|  |

* 1. Briefly what you think should be done at each stage for this calculator project.

|  |
| --- |
|  |

Once you have completed the task, you should **upload this document to EQUAL.**

## Task 2 (Week 2)

Your task is to create a prototype (early version) of the calculator program. You may create the program any way you wish, but it should meet the minimum requirements stated below.

*‘The program should allow the user to pick which mathematical calculation they want to perform (addition, subtraction, multiplication or division). Then, the program should allow the user to enter two numeric values, and then perform the chosen calculation, displaying the answer clearly to the user in a suitable way.’*

Add a screenshot of your code to your project document, along with screenshots to show your program working and performing all four arithmetic calculations.

|  |
| --- |
|  |

Once you have completed the task, you should **upload this document to EQUAL.**

## Task 3 (Week 3a)

Here is a list of roles commonly found in software development teams:

* Testers
* Project manager
* Designers
* Business analyst
* Software developers.

You have been asked to explain how these roles are involved in the calculator software development project.

To do this, you should complete the table on the next page, including the following for each role:

1. A brief description of what the role involves for this project (i.e. what/how would this person do?)
2. Who they would communicate with for this project
3. What they might deliver or produce for this project
4. What phase/phases of the SDLC they operate at.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Roles** | **What does the role involve?** | **Who might they communicate with  in a project?** | **What they might deliver or produce for this project** | **What phase/phases of the SDLC they operate at** |
| Testers |  |  |  |  |
| Project manager |  |  |  |  |
| Designers |  |  |  |  |
| Business analyst |  |  |  |  |
| Software developers |  |  |  |  |

Once you have completed the task, you should **upload this document to EQUAL.**

## Task 4 (Week 3b)

Return to your prototype of your calculator program, and add the following:

* Ensure that the program repeats (allowing the user to repeatedly perform arithmetic calculations without having to restart the program every time). The user should be able to end the program when they choose.
* Add comments to your code, to explain the code structure and elements.
* **Optional:** Add any additional functionality you wish, e.g., more advanced/scientific calculations, or the ability to do arithmetic with more than two values.

Add a screenshot of your code to your project document, showing the updated commented code and any necessary screenshots to prove the additional functionality you have added. Once you have completed the task, you should **upload this document to EQUAL.**

|  |
| --- |
|  |
|  |

# Unit 2 Project: Digital Processes and Services

### Project Overview:

In weeks 4-6, you will be gaining deeper insight into software development – including different ways in which software can be developed and tested, and the digital services and processes that are used by businesses.

### Project Scenario:

You have just joined a software development company as a trainee. The company makes software for educational establishments (schools, colleges and universities). As a new trainee, you have been given an introductory task – to create a simple calculator software program.

This software will eventually be used by primary school children (aged 4-11) to help them accurately learn their basic arithmetic skills (addition, subtraction, multiplication and division). It will also be a way for you to showcase your abilities to your new company and give them confidence in your ability to conduct all aspects of software development! You will have free reign to decide how the software will be designed and implemented, but it is suggested that the software is coded in a popular programming language.

Previously, you have already created a simple prototype to demonstrate the basic logic of the calculator program.

**Task 1 (Week 4)**

Up to this point, your company has been developing software in a traditional way, keeping lots of documentation and using a traditional waterfall (sequential) approach. You have been asked by your manager to investigate alternative methods.

1. Name some agile methodologies.
2. Describe what the key features of an agile methodology are, giving at least two advantages over a traditional waterfall/sequential approach.
3. Finally, state whether or not you feel an agile development approach would be suitable for the calculator project. Give at least one reason to justify your choice.

|  |  |  |
| --- | --- | --- |
| **Agile methodology** | **Key features of an agile methodology are, giving at least two advantages  over a traditional waterfall/sequential approach** | **Justification of why you think it would/would not be suitable for your project** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Once you have completed the task, you should **upload this document to EQUAL.**

## Task 2 (Week 5)

Currently all business resources in your organisation are stored on local servers and networks in the company’s head office building. All communication with customers is done via telephone and email. Your manager is looking at modernising the way the business works, to take advantage of digital web-based services and technologies.

You have been asked to add to your Task 1 document to cover the following:

1. List the common digital services that are provided by virtual web-based enterprises (minimum of four expected).
2. For the services you have listed, describe some of their features/functionality, and the technologies they might use.
3. Finally, of the services you have listed, explain which ones you would recommend your software development company could use and why.

|  |  |  |
| --- | --- | --- |
| **Common digital services  that are provided by virtual  web-based enterprises** | **Features/functionality, and the technologies they might use** | **Would you recommend your software development company could use? Why?** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Once you have completed the task, you should **upload this document to EQUAL.**

## Task 3 (Week 6)

While you still have more development work to do on your calculator program, now that you have a working prototype, it is a good idea to think about testing. Add the following to your document from the previous tasks:

1. Identify at least three types/levels of testing that would be suitable to perform for the calculator project, briefly explaining why the types of testing would be appropriate.

|  |
| --- |
|  |

1. Create a Test Plan table, and plan some test cases that would test your prototype calculator program and ensure all major functionality is working correctly. You are expected to use a variety of test data.
2. Execute your testing and update your Test Plan into a Test Log (by adding details and screenshot evidence of the actual test results). *\* Note: the expectation is that all four arithmetic calculations (+, -, \*, /) work correctly with various numeric data.*

Once you have completed the task, you should **upload this document to EQUAL.**

# Unit 3 Project: Development Practices

### Project Overview:

In weeks 7-9, you will be delving deeper into the best practice of coding, and examining the more advanced aspects of creating software.

### Project Scenario:

You have just joined a software development company as a trainee. The company makes software for educational establishments (schools, colleges and universities). As a new trainee, you have been given an introductory task – to create a simple calculator software program.

This software will eventually be used by primary school children (aged 4-11) to help them accurately learn their basic arithmetic skills (addition, subtraction, multiplication and division). It will also be a way for you to showcase your abilities to your new company and give them confidence in your ability to conduct all aspects of software development! You will have free reign to decide how the software will be designed and implemented, but it is suggested that the software is coded in a popular programming language.

Previously, you have already created and tested a simple prototype to demonstrate the basic logic of the calculator program.

**Task 1 (Week 7)**

After having recent training and discussions with your team about good coding practices, it is time to look at whether your prototype calculator program that you made previously is well coded or not. Create a new word document for this project.

1. Screenshot/paste in your commented code that you created previously for your calculator program.

|  |
| --- |
|  |

1. Explain whether your code follows ‘good coding practice’ for the following aspects:
   1. Comments (quality and quantity)
   2. Naming
   3. Layout
   4. Modularity and re-use.

|  |
| --- |
|  |

1. As a result of your evaluation, make any necessary improvements to your code, and screenshot/paste in your improved code. Ensure that improvements are clear and obvious.

|  |
| --- |
|  |

Once you have completed the task, you should **upload this document to EQUAL.**

## Task 2 (Week 8)

One thing that has cropped up recently with regards to the calculator project is that it turns out different schools would like different versions and configurations of the program – as an example, in the future, it is likely that there will need to be a version for very young children, and a more advanced calculator with different styling and arithmetic for older children.

To prepare for this, and to follow good practice in your organisation, your manager has recommended that you do the following:

1. Create an account at GitHub ( [https://github.com/](about:blank) ). Simply choose the free account type.
2. Create a new Repository (repo).
3. Ensure your repo has a ‘README’ that explain that the repo is for the calculator project.
4. Add the code for your calculator program so far into a new file your new repo.
   1. i.e. Create a new file in your repo and paste in your calculator program code. Add a description and commit your new file.

Take screenshots of your repo and file contents to prove the above is in place, and add these to your project document.

Once you have completed the task, you should **upload this document to EQUAL.**

|  |
| --- |
|  |

## Task 3 (Week 9)

More recent discussions about the calculator program have identified that the schools would definitely like a GUI version of the calculator, as well as a command-line version. In your project document, add the following:

1. Briefly discuss the key user interface differences between software based on desktop, web and mobile platforms. Which one would be best for the calculator project?

|  |
| --- |
|  |

1. Create a simple prototype GUI version of your calculator program. This is likely to have two textboxes in which to enter two numeric values, four buttons (one for each of the basic arithmetic calculations =, -, \*, /) and will display the calculation answer in a label. You can design/theme the interface how you like, but it should be appropriate for the project and take good UI practices into account.
2. Add a screenshot of your working GUI prototype calculator to your project document, along with all code.

Once you have completed the task, you should **upload this document to EQUAL.**

# Module 4 Project: Security and Databases

### Project Overview:

In weeks 10-12, you will be examining important aspects of software development that surround the core job of coding – namely security and storing data.

### Project Scenario:

You have just joined a software development company as a trainee. The company makes software for educational establishments (schools, colleges and universities). As a new trainee, you have been given an introductory task – to create a simple calculator software program.

This software will eventually be used by primary school children (aged 4-11) to help them accurately learn their basic arithmetic skills (addition, subtraction, multiplication and division). It will also be a way for you to showcase your abilities to your new company and give them confidence in your ability to conduct all aspects of software development! You will have free reign to decide how the software will be designed and implemented, but it is suggested that the software is coded in a popular programming language.

Previously, you have already created both a command-line and a GUI prototype to demonstrate the basic logic of the calculator program.

## Task 1 (Week 10)

After hearing that a rival software development company suffered a cyber security attack recently, management have decided that all staff need to be aware of the security dangers that exist in the industry. All staff have been given some development time to examine the security threats that are out there, and what can be done to mitigate these.

Create a new project document. Your first task is to use a table like that shown below to list and describe five IT security threats and how they might affect businesses (not individual home users). For each security threat, briefly describe how it works, what the impact could be, and how the security threat might be reduced or stopped.

|  |  |  |  |
| --- | --- | --- | --- |
| **Security threat** | **How does it work?** | **What could the impact be?** | **Mitigation / defence measures** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Once you have completed the table in your word document, you should **upload this document to EQUAL.**

## Task 2 (Week 11)

While the calculator project is not likely to involve a database, your manager knows that future development projects you might work on will often need a database to store the data needed by the software. To see if you have a grasp of the basics of relational databases, your manager has given you the following data:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Patient number** | **Patient name** | **Patient D.O.B** | **Patient gender** | **Patient phone** | **Doctor number** | **Doctor name** | **Doctor room number** |
| 55682 | Jeff Smith | 04/07/1979 | Male | 7876453 | 01 | Dr Hyde | 03 |
| 55683 | Olga Kuryleva | 08/02/1987 | Female | 2259985 | 02 | Dr Jekyll | 06 |
| 55684 | Samu Wat | 18/12/1999 | Male | 7971598 | 01 | Dr Hyde | 03 |
| 55685 | Lisa Elfton | 29/04/1993 | Female | 7071562 | 01 | Dr Hyde | 03 |
| 55686 | Rachel Green | 25/07/2003 | Female | 3691578 | 02 | Dr Jekyll | 06 |

Your manager would like you to analyse this data, and, in your project document, explain/show how this data should be effectively stored in a small relational database. For example, what tables, keys and relationships would you use? One important thing to note: in this data, it is expected that each patient might visit more than one doctor, and of course each doctor might see more than one patient.

Once you have completed the task, you should **upload this document to EQUAL.**

|  |
| --- |
|  |

## Task 3 (Week 12)

Schools have been shown the prototypes you have created previously, and early feedback is good. A couple of schools would like to know if the ‘memory’ functions that appear on a traditional handheld calculator can be created in some way.

Specifically, they have asked if you could look at adding the following functionality to the GUI prototype made previously:

* Pressing a button marked ‘M+’ will save the current value shown on the calculator to an external file
* Pressing a button marked ‘MRC’ will retrieve any value currently stored in the file, and put it in the first textbox (enabling it to be used in the next calculation)
* Pressing a button marked ‘MC’ will clear/delete any value stored in the external file.

Your task is to add these three buttons to your GUI calculator prototype, and add code to your program so that data values are stored to and retrieved from a file as described above. Any additional advanced memory functionality you wish to include will be welcome, but the above is the minimum required.

You should present some screenshots of your improved working program, and all program code in your project document. Once you have completed the task, you should **upload this document to EQUAL**

|  |
| --- |
|  |

# Congratulations!

You have now completed your Software Developer bootcamp project workbook.

Please ensure that you have uploaded the full booklet to EQUAL for your final assessment.

You should also keep a copy of your workbook for your own records as evidence of your work. This may be useful in the future when you are applying for jobs and experience in your chosen field.